II.2 Les sous-programmes

Comme vous le voyez, cela représente toute une masse de tâches secondaires. Et ces tâches secondaires doivent être à nouveau reprogrammées chaque fois pour tout programme. Pour chaque partie d'un programme doit être très nettement défini ce qui doit se passer dans l'ordinateur. Les choses sont encore compliquées par le fait que ces tâches doivent être résolues différemment pour les différents types d'ordinateur, aussi semblables qu'ils puissent paraître. Chaque ordinateur est en fait différent des autres. Si l'on écrivait donc d'une pièce un programme long, il devrait être entièrement réécrit pour chaque type d'ordinateur pour pouvoir fonctionner correctement.

On peut faire l'éconmie de ce travail inutile si l'on divise le programme en un programme principal et plusieurs sous-programmes. Les sous-programmes ne seront alors chargés dans la mémoire de travail de l'ordinateur que s'ils doivent effectivement être utilisés.

Pour adapter alors le programme à d'autres types d'ordinateur, il suffit de réécrire le ou les sous-programmes qui ne peuvent fonctionner sous cette forme sur le second ordinateur et la totalité du programme pourra tourner sans restriction sur le nouvel ordinateur. Un autre intérêt de cette technique des sous-programmes est que le programme principal ne sait plus du tout comment est résolue telle ou telle tâche parcellaire. Il se contente de donner une instruction, par exemple "sortir un texte" et le sous-programme résout ce travail. Que le texte soit sorti sur l'écran, sur une imprimante ou sur un télex n'a aucune importance pour le programme principal. Il donne simplement l'instruction et le travail est effectué.

Cette méthode fonctionne lorsque la transmission des données aux sous-programmes est standardisée dans le programme principal. On peut se représenter cela comme une course de relai où deux coureurs doivent toujours se transmettre le relai à un endroit très précisément fixé à l'avance. En langage informatique, un tel lieu

de transmission normalisé et déterminé précisément est appelé interface. Cette expression signifie qu'on peut à cet endroit retirer un sous-programme et le remplacer par un autre, sans que la fonction du programme principal n'ait à en souffrir. Tous les éléments restent adaptés les uns aux autres et fonctionnent sans problème.

Mais il est tout de même difficilement supportable pour chaque programmeur de devoir chaque fois réintégrer dans son programme les opérations internes. Cela représente beaucoup de travail et cela coûte beaucoup de temps et d'argent. C'est ce qu'on pensé également les hommes de DIGITAL RESEARCH qui ont donc eu l'idée d'écrire un programme standard pour les microordinateurs. Ils réunirent les programmes les plus courants qui sont nécessaires pour la gestion interne de l'ordinateur, ils définirent les interfaces et le mode de transmission des données entre programme principal et programme d'exploitation et ils réalisèrent ainsi un système d'exploitation pour les types d'ordinateur les plus différents.

Ce système d'exploitation ne peut bien sûr être utilisé que sur des ordinateurs disposant d'unités centrales semblables ou identiques car il faut bien que les instructions machine du programme puissent être comprises et traitées. Le premier système d'exploitation qui ait fourni de façon standard les nombreuses opérations de travail interne s'appelle CP/M. Cette abréviation signifie "Control Program for Micro-processors", soit programme de contrôle pour microprocesseurs. Ce système d'exploitation travaille avec les microprocesseurs du type 8080, 8085 ou Z80.

II.3 Les tâches de CP/M

CP/M résout pour l'essentiel les tâches de base suivantes: entrée de caractères, sortie de caractères, gestion de la place mémoire disponible sur la mémoire de masse, lecture des informations sur disquette et écriture de nouvelles informations sur disquette ou dans la mémoire de masse. Ces routines de service peuvent être utilisées par tous les programmes utilisateur d'une façon unique et